Multi-Screen Playback

Alright doggy - so you want to have multiple screens in sync? We know that we can get 7 @ 4k30 out of 16-Core i9 with 2x RTXA4000 cards. We know we can get the same out of an 8-Core i9 with 2x P4000 cards, but the performance sometimes dips. 2x Titan X cards also work, but it struggles a bit, too. Curious about what works? DIRECT Playback and mapping through MadMapper or playback through TOUCH and NDI'd/spouted to Mad for mapping.

Here are some build breakdowns and their achieved performance using MadMapper.

ID	CPU	GPU	RAM	Display Outputs (channels @ rez @ fps)
Krispy	Intel i9 8-Core / 16 Threads	2x P4000	64GB	7 @ 4k @ 30fps +1 Control @ 1080
Donna	Intel i9 16-Core / 24 Threads	2x RTXA4000	64GB	7 @ 4k @ 60fps +1 Control @ 1080
4x	AMD 24-Core / 48 Threads	4x RTXA4000	128 GB	10 @ 4k @ 60fps +1 Control @ 1080 (could probably handle more)

Here's a thread on MadMapper's forums describing the best workflow - again - who the F is this Trashcan guy??? He seems really smart.

FWiW - these builds are capable of channels x4 of 1080 in perfect sync if you split the 4k with something like an FX4 or HA5.

Breakout Example TouchDesigner @ 8 Outputs

6 @ 4k and 2 @ 1080p + Console Display

A summary of emails sent around by Sean Leo, Raphael Palefsky-Smith, Dave Tennent, Cam Vokey

The Good News

You can get 7 channels of 4k @ 60fps out of a 2x GPU system if the cards are P4000 or greater (newer flavor is the RTX A4000). Why 7 channels when there's 8 outputs? You need a control

monitor for your console, and if you use the on-board output, it drags your overall performance down. Technically this config can give you 28 displays with 7x fx4s (or HA5-4ks) – really cool. Obviously, you can draw headless (without a console) and have more screens, but I've found that if you can have a dedicated console, you should! Saves some pain down the road.

Mixed resolutions aren't your friend, but your total res is actually not bad. My recommendation is to check your performance and then combine your approaches so your computer is looking at only 4k displays. Your 6x 4k outputs are native through the GPU and your 2x 1080 outputs are through your FX in output #7 of the computer. Save output 8 for your console. Don't use the onboard video out (if you have one), it'll drag your performance down.

Performance Testing

First – you should find out if your computer is capable at playing at full resolution without being connected to external displays (make a TD window that's your target resolution total knowing you'll only see a portion of it). Watch your frame rate. If there's struggle here, your problem is your computer spec or your fx are too heavy. Or perhaps your level of TD license.

You can try pre-rendering some content at that resolution and playing it out using TD or MadMapper. In general, playback performance in MM is better than TD (and you get less tearing, etc). Also, MM is rentable.

What kind of content : render at your target resolution and embed some sort of timecode reference so you can visually see if you're dropping frames. Something that flashes every other frame is a good representation of performance drag.

You can also put a timecode number on every screen to check sync. Take a photo with your camera, if the number is the same (or even just between frames when you've got displays with different refresh rates), you're probably good.

If you have a TD Pro license, there's a special sync sauce at that license level. I've been in situations driving tons of 1080 displays with a comm license running – I plug in the Pro license, and the playback is suddenly perfect. If the media works in Mad, but not in TD, this should fix that problem. I'm not sure what the gatekeep feature here is, but it's pretty wild. Just checked and it seems you still can't rent a license :(

Once you get base performance to a level that meets expectations, read on.

GPU Outputs + FX4 Things

PCs (and Macs), while capable of playing out to a ton of displays, GPU bottleneck is created in a ton of different places, but the lowest lift is setting your computer to thinking it's using less displays than you actually are. The Datapath FX4 does this great as an external device, but your output resolution will be limited to 1080 per screen. If you're ok with that scale down on rez, then this is 100% a good call.

If you do explore the Datapath route, you may want to look into a sync generator and getting a sync card for the PC. This would help lock time between your GPU's, TD and the Datapaths.

Some GPUs allow you to create a FX4 like capabilities with the GPU config. On NVIDIA (non-gaming cards), this is called Mosaic. You can create a display that’s 7860x2160, for example. This will improve performance massively – drawing the same exact resolution to less displays, even though the rez is the same, is more efficient!

TouchDesigner Things

If using TD with Mosaic, you should look at GPU Affinity, which binds a TD process to a specific graphics card. The Derivative site has a good write up on the pros and cons of this method.. If you go that route, displays need to be matched, so use an FX4 for the two 1920 displays and a control monitor so the graphics cards are only looking at 4K displays.

One note from the TD programming side: The website recommends that you split the TD patches so one patch handles displays 1-4, patch 2 handles displays 3-7, and then both patches need to be launched via a BAT script. This would have some implications if you're doing content that wants to wipe across the entire canvas but that depends on the brief.

Other Methods to Improve Performance

Sometimes, sending media via Spout/Syphon or NDI from TD to MadMapper and then drawing to your displays there may actually improve performance. This is counter-intuitive : you’re doing more work. Mad is just really efficient at drawing to displays. Also, it’s a good place to crop and scale. If you need to add margins in between content here, or need to zhuzh some positions Mad is awesome at that.

Sometimes, having different display EDIDs brings performance down. Some headless passthrough (they fake an EDID) at 4k in all of your outputs may help.

Also you can try to reduce what your computer is doing – what is pre-render-able? What can be achieved through other means? Bake as much as the playback content as you can which will limit the amount of operations in TD, which will buy you some more overhead for performance.

Codec Stuff

The performance between NotchLC (NLC) and HAP on a TD system was fairly matched. The move was to pick the codec based on image quality. NLC was a bit better than OG HAP or HAPQ. HAPr (a newer proprietary flavor was recently improved by Jokyo) was better at gradients than NLC. The original HAP code included a version of HAPr (thanks Blair). But don't think you can encode / decode HAPr currently with FFMPEG.

You Can Also Try

Splitting the content across multiple systems and syncing via LAN. I avoid this as much as possible, but there are ways.

Additional Thoughts

TouchDesigner recently added the Direct Display Out TOP - a bit tricky to set up and you need Windows 11 IoT/Enterprise and a Quadro card, but it's really nifty. Instead of configuring a bunch of OS windows, placing them on the right monitor, hiding the mouse, worrying about DPI... you just send a texture directly to the monitor. It might make the setup harder for future maintainers - the monitors stop showing up in the Windows display settings, kinda confusing - but worth it? Sounds like it tricks your NVIDIA GPU into functioning like a BMD card.

Also: looking forward to the new Blackmagic 2110 card, which can send out 8x 4K streams or 32x HD over ST2110 Ethernet. So many pixels! You'd need a beefy 100G switch and a converter for each monitor. ST2110 is getting more and more traction.

Bottom Line

This resolution out of a properly configured single system (RAM, GPU, CPU etc) will work as long as the live rendering isn’t too complex. Find out what you can definitely achieve and then slowly add all your features back in until you get to your sweet spot!

Revision #5
Created 2025-04-14 18:42:58 UTC by Cam Vokey
Updated 2025-08-22 13:00:16 UTC by Cam Vokey